CLAIMS

1. A method for comparing a first order-independent data set comprising unique elements with a second order-independent data set comprising unique elements, the method comprising the steps of:

(a) for each entry in the first data set, placing the entry in a hash table;

(b) selecting an entry from the second data set;

(c) looking up selected entry in the hash table;

(d) removing, in response to locating the selected entry in the hash table, the selected entry from the hash table;

(e) determining if additional data set entries exist; and

(f) looping to step (b) in response to identifying additional second data set entries.

2. The method of claim 1 further comprising the step of identifying, in response to not locating the selected entry in the hash table, that the selected entry is second data set unique.

3. The method of claim 1 further comprising the step of performing, in response to not locating the selected entry in the hash table, a remedial function.

4. The method of claim 3 wherein the remedial function comprises deleting the selected entry of the second data set.

5. The method of claim 1 further comprising the step of identifying in response to no additional entries existing, any remaining entries in the hash table data as being first data set unique.

6. The method of claim 1 further comprising the step of performing in response to no additional entries existing, a remedial function.

7. The method of claim 6 wherein the remedial function comprises deleting the selected entry of the first data set.

8. The method of claim 6 wherein the remedial function comprises the step of transferring the selected entry from the first data set to the second data set.

9. The method of claim 1 wherein the step of removing the selected entry from the hash table occurs in response to identifying a match between a selected entry of the first data set and an entry of the second data set.

10. The method of claim 1 wherein the hash table comprises a B-tree.

11. The method of claim 1 wherein the hash table comprises a fast lookup data structure.

12. The method of claim 1 wherein the first data set comprises a set of directory entries on a source system.

13. The method of claim 1 wherein the second data set comprises a set of entries of a directory on a destination system.

14. The method of claim 1 wherein the first data set comprises a set of directory entries on a destination system.

15. The method of claim 1 wherein the second data set comprises directory entries on a source data set.

16. The method of claim 1 wherein first data set and second data set are on different storage devices.

17. A system for comparing a first data set with a second data set, the system comprising:

(a) means for placing each entry in a hash table;

(b) means for selecting an entry from the second data set;

(c) means for looking up the selected entry in the hash table;

(d) means for removing, in response to locating the selected entry in the hash table, the selected entry from the hash table;

(e) means for determining if additional data set entries exist; and

(f) means for looping to step (b) in response to identifying additional second data set entries.

18. The system of claim 17 wherein the hash table comprises a B-tree.

19. A computer readable medium, including program instructions executing on a computer, the program instructions including instructions for performing the steps of:

(a) for each entry in the first data set, placing the entry in a hash table;

(b) selecting an entry from the second data set;

(c) looking up selected entry in the hash table;

(d) removing, in response to locating the selected entry in the hash table, the selected entry from the hash table;

(e) determining if additional data set entries exist; and

(f) looping to step (b) in response to identifying additional second data set entries.

20. A method for comparing a first data set with a second data set, the method comprising the steps of:

creating a hash table of entries of the first data set;

locating, for each entry in the second data set, an entry in the hash table; and

removing, in response to locating an entry in the hash table, the located entry.

21. The method of claim 20 further comprising the step of recording, in response to not locating an entry in the hash table, that the entry in the second data set is second date set unique.

22. A method for comparing a first data set with a second data set, the method comprising the steps of:

(a) selecting an entry from the first data set;

(b) determining if the selected entry from the first data set is in a hash table;

(c) adding, in response to determining that the selected entry from the first data set is not in the hash table, the selected entry from the first data set to the hash table;

(d) removing from the hash table, in response to determining that the selected entry from the first data set is in the hash table, the selected entry from the first data set;

(e) selecting an entry from the second data set;

(f) determining if the selected entry from the second data set is in a hash table;

(g) adding, in response to determining that the selected entry from the second data set is not in the hash table, the selected entry from the second data set to the hash table;

(h) removing, in response to determining that the selected entry from the second data set is in the hash table, the selected entry from the second data set from the hash table; and

(i) independently continuing steps (a) through (d) and (e) through (h) for all entries in the first and second data sets until both the first and second data sets have been completely processed.

23. The method of claim 22 wherein the step of adding the selected entry from the first data set to the hash table further comprises the step of including information with the selected entry identifying the selected entry as originating from the first data set.

24. The method of claim 22 wherein the step of adding the selected entry from the second data set to the hash table further comprises the step of including information with the selected entry identifying the selected entry as originating from the second data set.

25. The method of claim 22 wherein the step of removing the selected entry from the second data set from the has table occurs in response to identifying a match between a selected entry from the second data set and an entry from the first data set.

26. The method of claim 22 further comprising the step of:

(j) recording all entries remaining in the hash table as being unique to either the first data set or the second data set.

27. The method of claim 22 wherein the hash table comprises a B-tree.

28. The method of claim 22 wherein the hash table comprises a fast lookup data structure.

29. The method of claim 22 wherein the first data set comprises a set of directory entries on a source system.

30. The method of claim 22 wherein the second data set comprises a set of directory entries on a destination system.

31. The method of claim 22 wherein the first data set and second data set are on different storage devices.

32. A system for performing a consistency check of a source directory replicated to a destination directory by comparing entries in the source and destination directories, the system comprising:

a process adapted to compare entries in the source directory with entries in the destination directory by walking the source and destination directories only once, whereby utilization of storage subsystems associated with the source and destination directories is limited by only walking each of the source and destination directories once.

33. The system of claim 32 wherein the process executes on a computer associated with the source directory.

34. The system of claim 32 wherein the process executes on a computer associated with the destination directory.

35. The system of claim 32 wherein the process is further adapted to remove matching entries from a hash table, whereby future look up operations in the hash table are enabled to be performed faster due to a smaller size of the hash table.

36. A system for performing a consistency check of a source directory and a destination directory by comparing entries in the source and destination directories, the system comprising:

a process adapted to select alternating entries from the source and destination directories to be added to a hash table and further adapted to remove matching entries from the hash table, whereby a size of the hash table is limited to a number of dissimilar entries of the source and destination directories.

37. A system for comparing entries in a source directory with entries on a destination directory to ensure consistency of replicated data between the source and destination directories, the system comprising:

a computer associated with at least one of the source and destination directories, the computer comprising a directory comparison process adapted to perform a comparison of entries in the source and destination directories by walking each directory once and placing entries in a hash table and further adapted to remove matching entries from a hash table, whereby computational cost is reduced for future look up operations in the hash table.

38. The system of claim 37 wherein the directory comparison process is further adapted to alternate in selecting entries from the source and destination directories when walking the source and destination directories.